

FAIRLIGHT

MUSIC COMPOSITION LANGUAGE



REFERENCE MANUAL

FAIRLIGHT MUSIC COMPOSITION LANGUAGE

M . C . L .

REFERENCE MANUAL

APRIL 1, 1981

© Copyright 1980 by Fairlight Instruments Pty. Ltd.

This documentation and associated programs are held copyright by Fairlight Instruments and may not be copied, altered, adapted, transferred or otherwise dealt with in whole or in part, without the express permission of the owners.

Fairlight Instruments Pty. Ltd.
15 Boundary Street,
Rushcutters Bay,
SYDNEY, AUSTRALIA 2011
Telephone (02) 33-5222
Telex AA27998

TABLE OF CONTENTS

Page

| | |
|---|----|
| SECTION 5 - M.C.L. EDITOR | 27 |
| 5.1 USING THE EDITOR | 27 |
| 5.2 COMMAND FORMAT | 28 |
| 5.3 FILE FORMAT | 29 |
| 5.4 CURSOR POSITIONING | 29 |
| 5.5 INPUT LINE | 30 |
| 5.6 ERROR MESSAGES | 30 |
| 5.7 COMMAND DESCRIPTION | 31 |
| 5.7.0 HOME (RESET CURSOR) | 31 |
| 5.7.1 ARROWS (MOVE CURSOR) | 31 |
| 5.7.2 RETURN (INSERT LINE) | 32 |
| 5.7.3 L-SUB (DELETE LINES) | 33 |
| 5.7.4 ADD (INSERT STRING) | 34 |
| 5.7.5 SET (OVERWRITE STRING) | 34 |
| 5.7.6 SUB (DELETE CHARACTERS) | 34 |
| 5.7.7 M (MUSIC KEYBOARD INPUT) | 35 |
| 5.7.8 L (LOCATE LINE NUMBER) | 37 |
| 5.7.9 S (RESEQUENCE LINE NUMBERS) | 37 |
| 5.7.10 D (DISPLAY TEXT) | 38 |
| 5.7.11 P (PRINT TEXT ON LINE PRINTER) | 38 |
| 5.7.12 F (FIND STRING) | 39 |
| 5.7.13 C (CHANGE STRING) | 39 |
| 5.7.14 \ (CHANGE PREVIOUSLY FOUND STRING) | 39 |
| 5.7.15 R (READ TEXT INTO COPY BUFFER) | 40 |
| 5.7.16 W (WRITE COPY BUFFER INTO FILE) | 40 |
| 5.7.17 B (OPEN COPY BUFFER) | 41 |
| 5.7.18 O (OPEN OR REOPEN FILE) | 41 |
| 5.7.19 N (CREATE NEW FILE) | 42 |
| 5.7.20 E (EXIT FROM EDITOR) | 42 |
| APPENDIX A - EXAMPLE SEQUENCE FILE | 43 |
| APPENDIX B - A SIMPLE EXAMPLE | 46 |
| APPENDIX C - EXTERNAL SYNCHRONISATION | 50 |

| | |
|---|----|
| SECTION 1 - THE M.C.L. CONCEPT | 1 |
| 1.0 STRUCTURE | 1 |
| 1.1 CAPACITY | 3 |
| 1.2 ACCURACY | 3 |
| 1.3 SYNCHRONISATION | 3 |
| 1.4 PRINCIPLE OF OPERATION | 4 |
| SECTION 2 M.C.L. FILES | 5 |
| 2.0 FILE STRUCTURE | 5 |
| 2.1 FILE MANAGEMENT | 5 |
| 2.2 SEQUENCE FILES: sequencefile.SS | 6 |
| 2.2.0 DEFAULTS | 6 |
| 2.2.1 NOTE SPECIFICATION | 8 |
| 2.2.2 REPEATS | 10 |
| 2.2.3 RELATIVE SPECIFICATION | 11 |
| 2.2.4 CONTROLS AND SWITCHES | 11 |
| 2.2.5 WAIT FUNCTION | 12 |
| 2.2.6 PROMPT | 13 |
| 2.3 PART FILE: partfilename.PT | 15 |
| 2.4 PIECE FILE: piecefilename.PC | 16 |
| SECTION 3 - M.C.L. COMMANDS | 17 |
| 3.1 LOAD,[filename] | 17 |
| 3.2 PLAY[,k] | 17 |
| 3.3 EDIT,[filename] | 18 |
| 3.4 NEW,<filename1>[,filename2] | 18 |
| 3.5 SAVE,<filename> | 18 |
| 3.6 CLEAR,<filename> | 18 |
| 3.7 DIR | 18 |
| 3.8 C | 19 |
| 3.9 RESET | 19 |
| 3.10 SPEED=<number> | 19 |
| 3.11 SYNC=INT | 19 |
| 3.12 CLICK=<rate>[,intrc] | 20 |
| 3.13 PRINT,filename | 20 |
| 3.14 COM | 20 |
| 3.15 WAIT=ON or OFF | 21 |
| 3.16 X=ON or OFF | 21 |
| 3.17 TIME=ON or OFF | 21 |
| 3.18 PROMPT=ON or OFF | 21 |
| 3.19 WARN=ON or OFF | 21 |
| 3.20 SYSTEM QUERIES | 21 |
| 3.21 DIV | 22 |
| SECTION 4 - DEBUGGING AIDS | 23 |
| 4.1 RUN-TIME ERRORS AND WARNINGS | 23 |
| 4.2 TIME FUNCTION | 24 |
| 4.3 HALT AND SINGLE-STEP | 25 |
| 4.4 "X" FUNCTION | 26 |

SECTION 1 - THE M.C.L. CONCEPT

-FAIRLIGHT M.C.L. (MUSIC COMPOSITION LANGUAGE)-

SECTION 1 - THE M.C.L. CONCEPT

1.0 STRUCTURE

M.C.L. is a TREE-STRUCTURED language. This means that a composition is treated as a "tree" containing three levels of hierarchy. These levels are called PIECE, PART and SEQUENCE. The tree structure means that a PIECE consists of one to eight PARTS to be played simultaneously, and each PART consists of one or more SEQUENCES which are played sequentially.

Although a maximum of eight parts can be played at once, the PIECE can contain a ninth PART as well. This PART can be used for programming control information, i.e. for varying the Page 7 controls and switches under M.C.L. control.

Sequences may be of any length from 1 to 2000 notes, although there is of course a limit to the total number of notes of sequence that can be accommodated at once. Note that more than one PART may share one or more common SEQUENCES. The one SEQUENCE can be played by several Parts simultaneously without any interaction between them.

It is convenient to think of each PART as a different "musician"; each "musician" plays a series of SEQUENCES on his own keyboard.

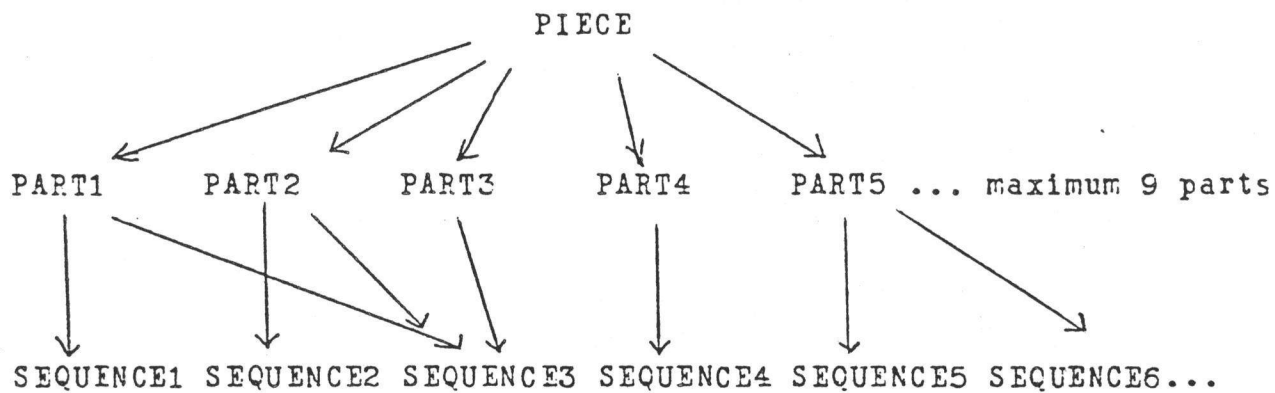
The playing of each PART is independent of other PARTS which may be playing at the same time.

The PIECE behaves like a "conductor", instructing which PARTS are required to play and when they are to start. Chords are permitted in SEQUENCES as long as the notes of the chord are all of the same duration and sufficient NPHONY has been allocated on page 3.

The PARTS may call for any number of sequences, with the restriction that a maximum of 32 files may be resident in the system at the same time, irrespective of their size. PARTS may also specify which conceptual keyboard is to be played by that PART, allowing switching from one register of the Fairlight to another between sequences.

SECTION 1 - THE M.C.L. CONCEPT (CONTINUED)

TREE STRUCTURE OF M.C.L.



SECTION 1 - THE M.C.L. CONCEPT (CONTINUED)

1.1 CAPACITY

Due to the enormous number of different possibilities available to the musician using M.C.L., it is meaningless to try to specify performance figures such as the maximum length of pieces possible or the number of keystrokes required to define a given amount of music. The answer to most such questions will generally be "that depends...". Suffice it to say that under WORST CASE conditions, that is without using any of the "tree structure" or other short-cuts, several minutes of complex polyphonic music can be held in the Fairlight's main memory. Using the full facilities it is a simple matter to specify hundreds of hours of music with just a few minutes typing.

1.2 ACCURACY

The time resolution of M.C.L. playing software is about 1 millisecond, so that very precise control of relative timing of events is possible. For example, rests of 1/100 of a beat can be used to generate subtle timing offsets to overcome the "mechanical" sound of a mathematically perfect performance. In fact, the timing control is sufficiently fine to achieve "flanging" effects by playing two identical voices slightly out of step.

1.3 SYNCHRONISATION

The Fairlight can accept an external SYNC input from which the M.C.L. replay system derives its timing reference. This can be used to synchronise to multi-track tape, motion-picture frames or any other source of pulses. By laying down a pilot track on tape before recording multi-part works, overall tempo variations common to all subsequent recording passes can easily be achieved.

It is also possible to synchronise a piece with any external event by use of the W function, which makes the M.C.L. halt at a predetermined place in the music until a key on the alphanumeric keyboard is struck.

SECTION 1 - THE M.C.L. CONCEPT (CONTINUED)

1.4 PRINCIPLE OF OPERATION

The M.C.L. system is supplied as part of a special Fairlight System Disk. The middle two digits of the version number displayed on Page 1 will be C followed by the version of M.C.L. on that disk.

Like the page 9 sequencer, the M.C.L. system plays music via Keyboards 1 to 8 on Page 3 of the Fairlight. After the required selection of voices and register allocations have been made on Page 3, the M.C.L. system is entered by typing PC <RTN> on any display page. The system will respond with the M.C.L. sign-on message, followed by the M.C.L. prompt. The prompt is the > symbol, and when it appears at the bottom left of the screen, it indicates that the M.C.L. is ready to accept a command.

Various commands are available for creating, editing, saving, loading and playing M.C.L. files. They are described in detail in section 3. Certain commands may result in a delay of a few seconds before they start execution, if requested immediately after page C is entered.

To enter a new composition, the M.C.L. Editor is used to create a Sequence File. This is done using the "NEW" command (described in section 3) to create the new file. The EDITOR commands (see section 5) are then used to enter the notes, durations and other musical data to create the Sequence file.

The data entered is not validated at this stage, that is you do not find out about errors until the M.C.L. tries to play what you have specified. When all the data for the sequence has been entered, the Edit is ended by typing E<esc>. The text disappears from the screen and the M.C.L. prompt appears again.

The sequence can then be played immediately using the PLAY command (see section 3). Any errors will be detected at run-time (i.e. as the M.C.L. tries to play what has been entered).

Invalid information in the sequence file will cause a WARNING message or an ERROR message to be displayed. A WARNING results when certain limits are exceeded, but the music goes on regardless. An ERROR results from feeding the M.C.L. data which it cannot understand at all, in which case it stops playing. Both types of message will display the offending line with an arrow pointing to the problem.

Further sequences can be entered in this way and played individually. Which sequence is played is determined by the sequence file-name which is entered as part of the PLAY command. To make a number of sequence file play one after the other, their names are entered into a PART file (see section 2), then the M.C.L. is instructed to play that Part. To play a number of sequences simultaneously, their names are entered into a PIECE file (see section 2) and the PIECE file is played.

The various files created are NOT automatically saved on disc. They are saved using the SAVE command (section 3) for later recall and editing.

SECTION 2 M.C.L. FILES

SECTION 2 M.C.L. FILES

2.0 FILE STRUCTURE

When creating or editing an M.C.L. file, line numbers are automatically inserted for the convenience of the musician. They are ignored by the system. Line numbering is useful for locating a particular point on a score if reference numbers are written down as the music is typed in. Run-time error messages print the line of text causing the error, with the line number of the offending line.

Comment lines may be included in any file. These must have an * in the first non-blank character position following the line number. Comments are for the musician's reference only and are ignored by the M.C.L. By placing an * at the beginning of any line of music, that line is turned into a comment and will therefore not be played.

All files must be terminated by an END statement. The END statement must be in a line of its own. Its purpose is to tell the M.C.L. that the end of the current file has been reached. Any text after the END statement will be ignored. Note that for obvious reasons, END may not be used as a filename.

Three different types of file are used by M.C.L. These are PIECE, PART and SEQUENCE files.

Extensive use is made of the concept of DEFAULTS. This is a time-saving technique whereby it is not necessary to specify every parameter controlling each event explicitly. For example, if a whole series of notes are to be played in the same octave, it is possible to set a DEFAULT octave which the M.C.L. will use whenever an octave number is not stated.

2.1 FILE MANAGEMENT

Composer file management is performed by the Fairlight PAGE 2 system. By requesting P2 and entering the command "M+", the displayed files will be those with .SS, .PT, and .PC suffixes. It is then possible to change the name of a file, delete a file or back up a file in the same way as normal Fairlight files.

SECTION 2 M.C.L. FILES (CONTINUED)

2.2 SEQUENCE FILES: sequencefile.SS

Sequence files are a list of the items to be played. These include default settings, note specifications, repeat directives, control parameter changes, relative operations and comments. The first non-blank character of each line in a sequence file has a special meaning:

* for comment lines (ignored)
! for default specification

Anything else means a line of music.
The following parameters may be specified as defaults:

2.2.0 DEFAULTS

BEAT- number of sub-divisions within each time unit, e.g. !B=16 means use default of 16 subdivisions per beat.

GAP- specifies the time between the end of the current note and the start of the next note. It is calculated in BEAT units. The GAP time must obviously not exceed the notes time value, and it must also be greater than zero.
!G=1/4 means the conceptual key-release occurs 1/4 of a "B" value before the next note starts.

HOLD- used as an alternative to GAP, specifies the duration (in B units) of the conceptual "key depression". HOLD time must be greater than zero and less than the note duration.

OCTAVE- specifies in which keyboard octave the specified note falls.
!O=2 means unless otherwise specified, notes are in octave 2 of the keyboard.

TRANSPPOSITION- adds an offset to the note requested. It is specified as the number of keys up or down the keyboard the note is to be moved.
!T=2 means play the note 2 keys (half tones) up the keyboard from where the specified note actually is. Transposition may be positive or negative.

VELOCITY- specify the effective "key velocity" used when playing a note. The key velocity data is used exactly as if it had originated from the music keyboard, it's precise effect being determined by PAGE 7 allocations. Velocity data must be in the range 0 to 15. !V=15 is the fastest velocity.

SECTION 2 M.C.L. FILES (CONTINUED)

KEY SELECTION- allows defaulting certain notes to be assumed sharp or flat to achieve a certain key signature. <cntrl s>, <cntrl f> and <cntrl d> are used on the alphanumeric keyboard to specify sharp, flat or natural respectively. !<cntrl S>F selects F to be sharp unless otherwise directed.

For example:

!<cntrl s>F:<cntrl s>C:<cntrl s>G sets key of A maj.

Several different default parameters may be specified within one line by separating them with colons, e.g.

!C=3:B=8:G=3/8:<cntrl f>B

SECTION 2 M.C.L. FILES (CONTINUED)

2.2.1 NOTE SPECIFICATION

Each note may be fully specified by:

pitch [velocity],[time],[gap or hold]

(Items shown in square brackets are optional; defaults apply if not supplied)

PITCH- The name of the note, A,B,C,D,E,F or G. It may optionally be preceded or followed by an accidental sharp, flat or natural which will override any key signature default currently in force.

The octave number may be specified immediately following the note (and accidental, if there is one). The octave number may be specified as relative to the default octave by using a + or - sign followed by the number of octaves above or below the default octave. E.g: A+2 means A two octaves above the default, so if an O=3 default had been set previously, the A will play in octave 5.

VELOCITY- The key velocity (as specified on page 7), can be changed for this note only. AV6 means to play note A with a key velocity of 6. (range: 0 to 15). Like the octave selection, a relative key velocity can be specified, e.g. AV+5 means use the default key velocity value plus 5 for this note.

TIME- The total time for this note (i.e. the time to the start of the next note rather than the time to the key release). It is expressed as a number of BEAT units, the absolute value of which will be determined by the SPEED selected. The TIME may be an integer or a fraction, and may be greater than one. The TIME value will be multiplied by the BEAT value to give the time to the next note, so it is important to ensure that the BEAT value and TIME are selected so that exact arithmetic is possible, e.g:

E=24 works with A,3/4 but with A,3/5 specified rounding errors would result in timing inaccuracies.

GAP- Time (in B units) between the conceptual "key release" and the start of the next note.

HOLD- may be specified instead of GAP time for any note or chord. This is the time (in Beats) that the note is to be held on. Thus either GAP or HOLD time may be specified for a particular note, or else a default value will be used. If a default is used then the last default specification will be used to determine whether the default gap or default hold time is used. Default hold time is specified by H=<number or fraction>. To specify a hold time for a particular note or chord the comma which normally precedes the GAP time is replaced with a fullstop (period). For example:

SECTION 2 M.C.L. FILES (CONTINUED)

A,1/2,1/4 ... note time 1/2, gap time 1/4

A,1/2.1/4 ... note time 1/2, hold time 1/4

(A:B),1/2.1/4 ... chord time 1/2, hold time 1/4

If there is no note time specified then the comma after the note name can be omitted altogether, thus:

A.1/4 ... note time 1, hold time 1/4

is the same as A,.1/4

also (A:B).1/4 ... chord time 1, hold time 1/4

If default gap is currently in effect and you want to use the default hold time then write the fullstop but with no time following, thus:

A,1/2. ... note time 1/2, hold time default

or A. ... note time 1, hold time default

Similarly when default hold is in effect, default gap can be forced by writing the comma with no time, thus:

A,, ... note time 1, gap time default

Note that forcing the other default value in the above cases will not affect the default value used for following notes. The general default is only changed by a G= or H=. The initial default assumed at the start of a piece is GAP.

Each note MUST have at least a PITCH specification. All other parameters for the note may default if desired. A REST may be implemented by specifying R in the PITCH field (instead of a note). Rests may be followed by a time value, but GAP and HOLD have no meaning in this case. If it is desired to default the time value but not the gap of a note or chord, double commas must be used between the pitch and the gap, e.g.

P,2,1/4 play B (may be sharp or flat depending on key selection), lasting 2 beats and released after 1 3/4 beats

B,,1/8 use default time value (i.e. same as B,1,1/8)

B4 Play B, octave 4 (i.e. override key signature)

<cntrl d>C4 Play C natural, octave 4

Several notes may be specified on the one line, separated by

SECTION 2 M.C.L. FILES (CONTINUED)

colons, e.g.

A:B:C:A,1/4:C,1/4,1/16

Notes to be played simultaneously (i.e. as chords) are grouped together in parentheses:

(A:E:A2:D2),3,1/2 : (A2:D2:D3:C)

In general, spaces in between note specifications are ignored, so that they can be included in the source text to make it more readable.

Appendix A gives examples of a variety of different note specification formats.

2.2.2 REPEATS

Repeats may be implemented at the sequence level by enclosing the section to be repeated in < > followed by the number of times the section is to be played. e.g.:

<A:B:C>3 means the same as A:B:C:A:B:C:A:B:C

Repeats may be nested up to six levels, e.g:

<<A:B:C>2 D>2 is the same as A:B:C:A:B:C:D:A:B:C:A:B:C:D

Repeats may spread over more than one line if required, but any repeat must be contained wholly within the one Sequence file.

SECTION 2 M.C.L. FILES (CONTINUED)

2.2.3 RELATIVE SPECIFICATION

All default values can be dynamically modified by specifying a RELATIVE value, that is to add or subtract some given number from the current value of the default, or divide or multiply it by some number. The format is similar to setting a default, except that a + or - is specified after the = to indicate a RELATIVE change, rather than an ABSOLUTE VALUE. For example: $O=+1$ will increment the octave number from whatever value it happens to be at that time. The effect of this is cumulative, so care is required to keep track of the current values, otherwise the permissible limits may be exceeded. If a value does exceed the limit, the M.C.L. will generate an error message when it attempts to play the sequence. Remember that the conventional default settings such as $O=3$ are absolute and will therefore always reset the default to the specified value.

It is also permissible to specify a relative ratio (ie multiply or divide by a constant). This is written as, for example, $B=*6$ or $B=/8$, where * represents multiplication and / represents division. This method of changing speed enables different sequence files to have different B values and still keep in step when a speed change is desired. However the remarks made earlier about ensuring that the numbers are compatible to avoid rounding errors also apply here. Thus if the current value of V is 10 and the command $V=/7$ is given then a rounding error will result.

2.2.4 CONTROLS AND SWITCHES

The PAGE 7 controls and switches can be specified from an MCL sequence by specifying:

and $Cn=<\text{number or relative specification}>$
 $Sn=0$ or 1 or OFF or ON

These may appear anywhere in a sequence file (except on a comment line) including a default specification line (preceded by a !). For example:

! $C1=120 : S2=ON : S3=0$

* NOTE THAT $C1=120:S2=1:S3=OFF$ WOULD DO THE SAME

* CONTROL 1 SET TO 120; SWITCH 2 ON; SWITCH 3 OFF

.

.

.

$C1=/3 : C1=*2$

* CONTROL 1 IS DIVIDED BY 3 AND MULTIPLIED BY 2, GIVING 80

SECTION 2 M.C.L. FILES (CONTINUED)

The M.C.L. keeps track of control settings for as long as a piece is in progress. The M.C.L. controls and switches will override the actual switches, controls and pedals on the music keyboard as long as these are not operated at the same time. The range of values permitted for controls is 0 to 127. If this range is exceeded then the value will be limited, and a warning given (if warning messages are enabled). When relative specifications are used bear in mind that if more than one sequence references the same control then the sequences may interact in an unpredictable manner. Controls and switches (along with all other sequence file specifications) may be used in part files if the line contains a " in the first column.

2.2.5 WAIT FUNCTION

By inserting the character W into a sequence file the music can be synchronised with any external event. If the WAIT function is enabled (by WAIT=ON), then when the PLAY command reaches the W, the playing of music will be suspended until any key on the alphanumeric keyboard or music keyboard keypad is struck.

The key may be struck in advance so that there is no wait at all. As soon as the key has been struck, the music will then resume as if nothing had happened.

SECTION 2 M.C.L. FILES (CONTINUED)

2.2.6 PROMPT

This feature allows a piece of music being played by M.C.L. to give cues to the musician by displaying pre-programmed messages on the music keyboard display, or the graphics screen. It is also possible to send data to another computer or synthesizer which can be connected to the CMI's line printer output or parallel interface output. The commands for enabling this function and determining which output device will be used are described in section 3.

An output message (or control data) can be inserted in any sequence file (or after a " in the first column of a part file) by enclosing the string of characters in square brackets [] (these are obtained by typing shift-K for [and shift-M for]). The string must be entirely on one line of the file. Any printable character may be sent, and by using the special character : (colon), any unprintable character may also be sent. The colon indicates either a hexadecimal value or a control character definition. If the character following the colon is a valid hex digit (ie in the range 0 to 9 or A to F) then it and the following character are converted to a hexadecimal number. If the second character is not a valid hex digit then the prompt string will be terminated. The first character after the colon may also be one of the following special control characters:

R,L,P,N,S,X,H which will cause the corresponding control character from the table below to be sent:

| | | | |
|----|-----------|-------------------|--------|
| :R | will send | <carriage return> | hex 0D |
| :L | will send | <line feed> | hex 0A |
| :P | will send | <form feed> | hex 0C |
| :N | will send | <null> | hex 00 |
| :S | will send | <escape> | hex 1B |
| :X | will send | <backspace> | hex 08 |
| :H | will send | <cancel> | hex 18 |

In these cases only the first character after the colon is treated as a special character. It is not necessary to use any of these control characters when sending messages to the music keyboard display. The other special character, " (double quote), forces the following character to be transmitted exactly as is. Thus to send a colon use ":" to send a right square bracket use "]" and to send a double quote use "\". EXAMPLES:

| | | |
|----------------|-------|----------------------|
| [ABCDEFGH] | sends | ABCDEFGH |
| [ABCDEFGH:R:L] | sends | ABCDEFGH<cr><lf> |
| [:7F"]":":0A] | sends | <rubout>]:<linefeed> |

The timing is arranged so that the transmission of the string will begin at exactly the time specified in the sequence file (provided that the output device is ready), and will continue as a background task to the playing of music. Thus the time taken to output the string will not affect the timing of the music in any

SECTION 2 M.C.L. FILES (CONTINUED)

way. If a second literal output request is encountered before the previous one is completed then the first one will be truncated and the next one will start immediately.

SECTION 2 M.C.L. FILES (CONTINUED)

2.3 PART FILE: partfilename.PT

PART files are a list of SEQUENCE FILES to be played sequentially by that PART. Keyboard number played may also be set here, using IK=n as a line in the file. In the following example the sequences SEQ1 and SEQ2 will be played on keyboard #1 and sequences SEQ3 and onwards will be keyboard #2.

It is also possible to insert lines of SEQUENCE file information (i.e. notes, rests, chords and default information) into a part file by placing a double quote character (") in the first position in the line. Whatever follows this character will be treated exactly as though it was within a sequence file. Thus, if it is necessary for a PART to play a SEQUENCE, then wait a while before playing the next SEQUENCE, the required number of rests can be inserted in the PART file.

PART files are of the form:

```
* This is a comment.
* START PLAYING KEYBOARD #1
IK=1
SEQ1.SS
SEQ2.SS
* NOW SWITCH TO KEYBOARD #2
IK=2
SEQ3.SS
* WAIT FOR 4 BEATS, THEN PLAY SEQ1.SS AGAIN
*
* THE DOUBLE QUOTE WILL MAKE THE FOLLOWING LINE INTO A SEQUENCE
"R,4
SEQ1
.
.
.
END
```

Note that as only SEQUENCE names may appear in PART files, the .SS suffix is redundant and therefore optional.

SECTION 2 M.C.L. FILES (CONTINUED)

2.4 PIECE FILE: piecefilename.PC

PIECE files are a list of PARTS which are to be played simultaneously. The order in which the PART files are specified determines the keyboard number that part will play unless an over-riding IK= statement is used in the PART file. In the following example, PARTA will play keyboard #1, PARTB will play keyboard #2 etc.

PIECE files are of the form:

* This is a comment.

PARTA.PT

PARTB

PARTC

SEQ1.SS

.

.

.

END

In general PIECE files will contain only PART files, so the .PT suffix on filenames is optional and may be omitted. However PIECE files may contain SEQUENCE filenames provided that the suffix .SS is explicitly stated.

SECTION 3 - M.C.L. COMMANDS

SECTION 3 - M.C.L. COMMANDS

A number of commands are available for selecting various M.C.L. functions such as editing, loading, saving or playing files. Commands may only be entered when the prompt > is displayed on the screen. This means that a command can not be entered while a PLAY is in progress, which must be terminated with <cntrl ESC>, or an EDIT is in progress, which must be terminated by <E ESC>.

Some commands require a filename to specify which file is to be acted upon. A default system is provided in some cases, whereby if no filename is specified the name last used will be assumed.

Certain commands may result in a delay of a few seconds before they start execution if requested immediately after page C is entered.

Filenames must always be specified with a file suffix of:

.SS for Sequence files
.PT for Part files
.PC for Piece files

Filenames may be any valid Fairlight-type name (8 characters maximum, first character must be alphabetic).

Commands are:

3.1 LOAD,[filename] or L,[filename]

Load a file from disc into memory.

The contents of a file may be used to invoke loading of further files by enclosing the filename in () e.g.
LOAD (PART.PT) will load the part file PART, then read the file to see which sequences are called for by that part. These are in turn loaded.

LOAD ((PIECE.PC)) will load the piece, then the part, then the sequences within the parts.

If no filename is specified, the LOAD command defaults to the last name specified for a LOAD.

The letter L on its own can be used in place of the word LOAD for convenience.

3.2 PLAY[,k] or P[,k],[filename,filename,filename,...]

Play all files specified simultaneously. The files may be a mixture of PART and SEQUENCE files. [k] is an optional Keyboard Number. The files specified will be assigned to successive keyboards starting with keyboard "k" if specified. If no "k" is specified, the first file will be assigned to keyboard one, the next to keyboard two, and so on.

Alternatively, a single PIECE filename may be specified.

SECTION 3 - M.C.L. COMMANDS (CONTINUED)

The letter P on its own may be used instead of the word PLAY.

If no filenames are specified with the PLAY command, the names used for the last PLAY will be used. If it is the first time a PLAY has been specified since system start-up, or since the last RESET command, then the last filename loaded will be assumed. If no file has been loaded since the last RESET then the last file edited will be played.

To abort a PLAY, type <cntrl ESC>.

3.3 EDIT,[filename] or E,[filename]

Edit a file. Filename must be a currently loaded file. Whenever the editor opens a file for editing it is first checked to ensure that it is a valid M.C.L. file. This means that it must contain correctly formatted sequential line numbers, all lines must be less than 80 characters long and it must end with a <carriage return>. Any file which does not meet these criteria will be rejected and the message: "BAD FILE, CANNOT EDIT" will be displayed.

Refer to Section 3 for details of Editor commands.

3.4 NEW,<filename1>[,filename2]

Create a new M.C.L. file. The file will be created in memory, but no disk allocation is made. <filename> must not already be loaded into memory.

If filename2 is specified, the file .created will be initialised to be a copy of filename2.

3.5 SAVE,<filename>

Save a file to disc. If the file specified is already on the disk it will be overwritten. The same format as load is allowed, i.e. SAVE ((filename.PC)) will save the piece, parts and sequences.

3.6 CLEAR,<filename>

Remove the file specified from memory. This makes the memory and directory space occupied by that file free but it DOES NOT save the file to disc before clearing it.

CLEAR,* may be used to clear all files. It also clears the default PLAY and EDIT filenames.

3.7 DIR

Display list of filenames currently loaded. If <space>A is specified after the DIR command, the SIZE of each file (in bytes) will be displayed next to its name.

SECTION 3 - M.C.L. COMMANDS (CONTINUED)

3.8 Q

This is the QUERY command, which displays the state of various system variables, i.e.:

SPEED, SYNC, CLICK, X, TIME, WAIT, WARN, PROMPT, and FREE (memory unused).

3.9 RESET

Reset the values of all system variables (see above under Q command) to their defaults. These are as follows:

| | |
|--------|-----------------------------------|
| SPEED | 1000 |
| SYNC | INT (EXT division ratio set to 1) |
| CLICK | OFF 48,4 |
| X | OFF |
| TIME | OFF |
| WAIT | OFF |
| WARN | OFF |
| PROMPT | OFF |

3.10 SPEED=<number> or S=<number>

Set replay clock speed in microseconds per tick.
i.e. SPEED=10000 means 10 ms. / tick at which speed
using B=24 a note of time-value 1 will last 240 ms.

The default speed on system start-up is 10000.
The permissible range of values for the SPEED command is
1000 to 60000.

The letter S on its own can be used instead of the word
SPEED.

3.11 SYNC=INT or EXT or number

Select the M.C.L. replay synchronisation mode.

INT means use internal speed setting.

EXT means use SYNC INPUT pulses as clock. Two sync
pulses = one clock tick ("B" unit)

"number" means use external sync divided by the number.

The permissible range of values for external sync is
1 to 60000.

SECTION 3 - M.C.L. COMMANDS (CONTINUED)

3.12 CLICK=<rate>[,intro]

Means generate a click track from the sync output and/or monitor amplifier. "Rate" is specified in BEAT units, and "Intro" is the number of clicks to be issued before the playing starts. The CLICK volume heard in the monitor speaker can be adjusted using the SYNC MONITOR control on the rear panel of the Fairlight.

The range of values allowed for rate and intro is 0 to 255. If no value is given for intro then the number 4 will be assumed.

CLICK=CN and CLICK=OFF can be used to turn the CLICK function on and off.

3.13 PRINT,filename

Print the file specified on the line printer. The SHARP, FLAT and NATURAL symbols will be printed as s, f, and n. The same format as the LOAD command can be used to print parts and sequences automatically, i.e. PRINT ((filename.PC)) will print the PIECE, PARTS and SEQUENCES.

3.14 COM

The COM (COMPILE) command has been provided to enable M.C.L. pieces to be converted to sequencer files so they can be replayed or merged on PAGE 9. The format of the COMPILE command is as follows:

COM[,output filename][,keyboard] [input filename(s)]

All the parameters are identical to the PLAY command except for <output filename>. This is the sequence file which is to be created and must have the suffix .SQ or no suffix at all.

If the output file already exists then you will be asked whether you want to overwrite the existing file. Any response other than Y<cr> will terminate the compile command.

If a COMPILE has already been done then the previous output filename will be assumed if none is specified. The input filename can be any M.C.L. file or list of up to eight sequence files and part files. If no input filename(s) are specified then the same default name as would have been used by the PLAY command will be taken. If an input filename is specified then the default PLAY filename will be changed to the name specified in the COM command. If the keyboard number is to be specified but no output filename then only one comma should be used between COM and the number.

SECTION 3 - M.C.L. COMMANDS (CONTINUED)

3.15 WAIT=ON or OFF

Turn the WAIT function on or off (see section 2.24)

3.16 X=ON or OFF

Turn the X function on or off (see section 4.4)

3.17 TIME=ON or OFF

Turn the TIME function on or off (see section 4.2)

3.18 PROMPT=ON or OFF

Turn the PROMPT function on or off (see section 2.2.6)

3.19 WARN=ON or OFF

Turn the WARNING on or off (see section 4.1)

3.20 SYSTEM QUERIES

To find out the current settings for SPEED, SYNC, CLICK, TIME, WAIT PROMPT and X type in the name of the variable followed immediately by a carriage return. The system will respond by displaying the current value on the graphics screen.

SECTION 3 - M.C.L. COMMANDS (CONTINUED)

3.21 DIV

Calculate the number of beats corresponding to a certain number of clock ticks at a given B value. The command format is:

DIV,<ticks>,<B value>

The output produced is as follows:

<num>/<denom> OR <quot> + <rem>/<denom>

Where <num> and <denom> are the improper fraction corresponding to the required number of beats (ie same as <ticks>/<B value> but reduced to lowest terms), <quot> is the integer quotient of <num> and <denom>, and <rem> is the remainder.

This command is intended for converting the numbers of clock ticks output by the TIME (?) function.

For example, assume that a ?request informs you that at a certain point in a sequence 1000 clock ticks have elapsed. You want to synchronise a second sequence so that it starts playing at this point; the second sequence has a B value of 36.

How many rests are required?

Use the DIV command as follows:

DIV,1000,36

The following output will result:

250/9 OR 27 + 7/9

Thus you can either use R,250/9 or R,27:R,7/9 to produce the required delay. It is generally preferable to use the second format as the first method stands a greater chance of producing an arithmetic overflow.

SECTION 4 - DEBUGGING AIDS

SECTION 4 - DEBUGGING AIDS

4.1 RUN-TIME ERRORS AND WARNINGS

At Run-time (that is when the PLAY command is executed) the Composer system starts interpreting the PIECES, PARTS or SEQUENCES and executing the performance according to the data it encounters. At this time various types of invalid data may be detected, and appropriate messages will appear on the screen.

The M.C.L. differentiates between two levels of severity of such problems by classifying them as WARNINGS or ERRORS.

Warning messages are given for some common limiting conditions. These are:

- a) Octave value out of range (not in range 1-7)
- b) Velocity value out of range (not in range 0 to 15)
- c) Gap time (or Hold time) of zero
- d) Gap time (or Hold time) longer than the note
- e) Control value (see above) out of range (not in range 0-127)

When an out-of-range condition occurs the value will be forced to the limiting value which was exceeded. In the case of gap and hold time, if the value calculated is zero then a value of 1 clock tick will be used. If the value is greater than or equal to the note time then a value of note time minus 1 clock tick will be used. Warning messages are displayed in the same format as error messages but the associated part will not be terminated and playing will continue as normal. However if the warning appears very near the start of a "play" there may be a delay (and some lost clock ticks) because the message requires a disk access to be made. When the system is first loaded warning messages are disabled (i.e. the limiting condition is handled as above, but no message appears and no time will be lost by disk access); to enable messages for debugging the command WARN ON (or WARN=ON) should be given. After a piece is debugged then the warning messages can be disabled by the command WARN OFF (or WARN=OFF).

ERRORS result from :

- a) PARTS or PIECES requesting files that are not loaded.
- b) Syntax errors in the files.
- c) Any characters in a file which is M.C.L. does not recognise.

When an ERROR is detected, the PART concerned will stop playing, whereas after a WARNING it will continue.

All messages are accompanied by the name of the file in which the error occurred. The offending line and its line number will be displayed, and an upward arrow will be displayed underneath the first character in the line which seemed to be responsible for the error. In some cases this will point exactly to the error, in other cases it will only provide a clue.

SECTION 4 - DEBUGGING AIDS (CONTINUED)

Note that the interpretation of composer files is always some time ahead of the music you hear, so that error messages will usually appear on the screen before the music being played reaches that point.

If, after a piece has finished playing, the message

"WARNING: CLOCK TICKS LOST = nnnnn"

appears, this indicates that the speed at which the music was played was so high that the composer system was not able to interpret the files fast enough to keep up with the rate at which the music was being played. "nnnnn" indicates how many B units were spent waiting for the composer to catch up. This condition should never occur in normal operation, but may appear if the X function is used to turn off the music in the middle of a piece or if a lot of WARNING messages were being generated.

4.2 TIME FUNCTION

When creating a piece of music using the composer it is often necessary to calculate the number of beats up to a certain point in the music, or between two points, in order to synchronise with another part.

The TIME function (in conjunction with the X function described below) can be used to ease this problem. By inserting a question mark at a certain point in any sequence file, the computer will, when it reaches the question mark during a PLAY command, display the following data:

```
FILE = <seq>/<part>/<kbd> TIME = <time1>/<time2>  
      <line of sequence containing the ? >
```

↑
<pointer to exact position of the ?>

Where: <seq> is the name of the sequence
 <part> is the name of the part playing the sequence
 <kbd> is the keyboard number being played
 <time1> is the time since the last ?
 <time2> is the time since the start

As an example, the number of rests needed to make a new sequence come in at a particular point is obtained by dividing the number of clock ticks up to that point (as shown by the TIME function) by the value of B.

The DIV command can be used to calculate correct durations from given Time and Beat values (see section 3.21).

SECTION 4 - DEBUGGING AIDS (CONTINUED)

4.3 HALT AND SINGLE-STEP

It is possible to interrupt a piece of music at any time by pressing the CTRL-W key (ie press the W key while holding the CTRL key down). This will have exactly the same effect as if a WAIT command (W) had been encountered in the piece being played, except that it will function regardless of whether the WAIT feature is enabled or not. To resume playing press any key on the keyboard. It is suggested that a non-printing key (such as HOME, CLEAR, ESC or an arrow) is used so that it does not appear when the piece is finished.

The single step feature is an extension of this. If the music is halted (by the CTRL-W key or the WAIT function) and then the CTRL-W key is pressed to restart the playing, the system will advance to the next time unit (ie a key depression or release) and then stop again. This process can be continued indefinitely. Note that since, in general, each note that you hear is two time units (a depression and a release), the system will play one note or chord for each two CTRL-Ws.

SECTION 4 - DEBUGGING AIDS (CONTINUED)

4.4 "X" FUNCTION

This function enables the user to selectively omit certain parts of a piece when the PLAY command is used, while still keeping all parts in correct synchronisation with each other.

By inserting the character X in a sequence file, followed by a plus (+) or minus (-) sign the user can turn the actual playing of the piece on and off. The character string "X-" will turn the music off, while "X+" will turn it back on. Note that the COMPOSER still has to read all of the music, in order to keep track of default values, timing etc, but no music will be played. Because of this, there may be a delay of a few seconds (or even a minute if it is a very long piece) before the music resumes, while the Composer skips through the files looking for the next X+.

Redundant X+ and X-'s (eg X- when the music is already off) will be ignored. This feature is especially useful when the musician is working on the end of a long piece, as it enables him to miss out the parts which have already been corrected, and just listen to the part he is working on.

Both the X and the ? features can be used in PART files by making use of the QUOTE feature ("). (see under PART files). Thus to listen to the last sequence only in a part, and to find out how many ticks into the piece it starts, the following construct may be used in the PART file:

```
* TURN THE MUSIC OFF
"X-
SEQ1.SS
SEQ2.SS
SEQ3.SS
* NOW TURN IT BACK ON AND FIND OUT HOW
* MANY TICKS HAVE ELAPSED
"X+ ?
SEQ4.SS
END
```

When the X and TIME functions are disabled by the respective =OFF commands they are ignored by the composer and may be left in the files if desired.

SECTION 5 - M.C.L. EDITOR

SECTION 5 - M.C.L. EDITOR

5.1 USING THE EDITOR

The EDITOR is used to create or modify M.C.L. files. It includes many powerful features to enable rapid text generation with a minimum of typing effort.

The editor is invoked by means of commands entered after the M.C.L. prompt appears on Page C. For example, to create a new Sequence file called "SOLO", enter the command:

```
NEW,SOLO.SS
```

Remember, all M.C.L. commands (but not all EDITOR commands) must be terminated by the RETURN key <RTN>.

The screen will then change its format, and you will be presented with an almost empty display. The second line from the top of the screen will echo whatever characters you type. Nothing else will happen until you issue an EDITOR command. To do this, it is necessary to indicate that you have finished entering the text data and you want the Editor to perform some action, such as inserting the line of text into a file.

For this purpose certain control characters are used as commands. A control character is not printable, so it can not be confused with part of the text. The functions of the various control characters is described in detail in the remainder of section 5.

As text is inserted into the file, it appears at the middle line of the screen, and line numbers are automatically generated for reference purposes.

When the Editing is complete, the command <E ESC> is used to return to the M.C.L. system (i.e. type an E followed by the ESC key).

The command "NEW" is used to create a new file, the command "EDIT" is used to modify an existing file. The file to be edited has to be loaded from disk before the EDIT command is given.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.2 COMMAND FORMAT

Note: In the following descriptions, the word "string" means any group of alpha-numeric characters, which may include spaces. The < xxx > notation is used to indicate a special control key.

Commands take two general formats. The first consists of a string terminated by a control character which defines the command; the second consists of a string containing a command letter and terminated by the control char <esc>.

Commands of the first type are:

<home>,<up arrow>,<down arrow>,<left arrow>
<right arrow>,<RTN>,<add>,<sub>

Commands of the second type are:

M,L,S,D,P,F,C,\,R,W,B,O,N,E

Commands of the first type have no fixed format, the other commands all have the structure:
<number(s)><command letter><string(s)><esc> Multiple numbers must be separated by commas.

If any optional numbers or strings are not supplied, default values will be assumed. Those commands which do not require strings will give a syntax error if the command letter is not the last character in the input line.

For some commands the string must be a valid filename. Filenames must always have a valid M.C.L. file suffix:
.SS, .PT or .PC .

Numbers or filenames may have any number of spaces on either side, and type 2 commands always ignore any leading spaces. Numbers may in almost every case be replaced by the special character "&" which is considered to have a value of 10,000. All numbers must be four decimal digits or less, and may be preceded by a plus or minus sign.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.3 FILE FORMAT

M.C.L. files have line numbers attached to every line. The line number is an integral part of the line and cannot be deleted unless the whole line is deleted.

Line numbers are in the range of 0000 to 9999 and must be sequential; although the editor will allow adjacent lines to have the same number they may not decrease in value.

If a line is created which has the same number as the one before or after then the warning message "Line Number Conflict" will be given and it is recommended that the "resequence" command be used to remove the conflict before ending the edit. This will avoid confusion later as line numbers are assigned automatically by the editor when a line is inserted.

The maximum length of any line created by the editor cannot exceed 79 characters (not including line number).

5.4 CURSOR POSITIONING

While editing a file the current character position (cursor) being edited is defined by:

- 1) the current line and
- 2) the character position within the current line.

Initially the cursor will be at the home position (beginning) of the first line of the file.

When at the home position the cursor is pointing at the line number and this is indicated by the line number being displayed as black-on-white on the screen. In this position lines of text can be inserted sequentially into the file in front of the current line.

When the cursor is moved out of the home position by use of the right or left arrow commands, or by a character string search or change command, then the character being referenced is indicated by a break in the window surrounding the current line either side of the character. The actual point of edit is immediately to the left of this character.

When the cursor is not at the home position then the line oriented commands (RETURN, L-SUB, R, W, S) will behave differently. The cursor can be moved back to the home position at any time by use of the HOME command.

If the cursor is moved to the end of the file then the current line will be empty (ie no line number and no text). When text is added a new line number will automatically be assigned.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.5 INPUT LINE

Characters are entered into the command line and are not processed until a control character is received. There are two special control characters which operate only on the command line. These are <clear> and <cntrl B>.

The CLEAR command will erase the current command line as on any other page of the Fairlight.

The <control-B> command, which will be referred to as RESTORE, causes the command line to be reset to its condition at the time of the last editor command.

Every time a command is executed the command line is saved. (there are some exceptions to this.... see below) If the RESTORE command is issued, characters which have been typed in since the last command will overwrite the previous contents of the command line, but the rest of the command is not changed. This feature is very useful if, for example, an error was made right at the start of a command. The first part of the line can be retyped, and then <cntrl B> is hit to RESTORE the rest of the line, saving a lot of typing.

The command line is cleared after each command has finished execution except for the following cases:

- a) after an error condition occurs
- b) the command was a simple cursor movement

RESTORE can not be used following commands:

D<esc>, <esc>, <home>, <sub>, all arrow commands.

5.6 ERROR MESSAGES

Error messages are printed on the top line of the screen and are cleared as soon as the next input character is received (unless its <clear>, which only clears the input line)

If an error occurs due to bad syntax of the command line (ie. a typing error or other mistake by the user) then the command will not be executed at all. If the error occurs for some other reason then the command may or may not be executed, or only partially executed, depending on the nature of the error.

For multiple repetition commands the message "warning" and "count=nnnn" may appear, indicating that the number of repeats requested could not be executed. In this case nnnn represents the number of repetitions that were actually made.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7 COMMAND DESCRIPTION

5.7.0 HOME (RESET CURSOR)

<home>

Move the cursor to the home position. If already there then do nothing. The input line is not affected and may contain anything. There are no error conditions for this command.

5.7.1 ARROWS (MOVE CURSOR)

<right,left arrow>

Move the cursor right or left. If at home then left arrow has no effect, if already at end of line then right arrow has no effect.

<up arrow>

Move cursor up (ie towards start of file). Thus the text on the screen will scroll downwards.

<down arrow>

Move cursor down (ie towards end of file). Text will scroll up on screen. The cursor may move past the last line, which is known as the "end-of-file" position. As this line is completely empty no line number will appear until something is inserted into the line.

For the up and down arrow commands the cursor will attempt to remain at the same character position in the new line; however if the new line is not long enough then the cursor will be positioned at the end of the new line. For the multiple line moves (see below), the cursor position is not affected by the lengths of intervening lines, only the starting and final lines.

All of the four arrow commands may be preceded by a repetition number providing that it is the only thing in the input line. The infinity character (&) can also be used. However if there is more than a single number, or extra characters in the input line then the arrow commands will always take a default value of 1. The same applies if an invalid number is found (ie too many digits). The input line is always unchanged by cursor moving commands.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

TEXT INSERTION AND DELETION COMMANDS

5.7.2 RETURN (INSERT LINE)

<string><carriage return>

This command inserts a line of text into the file at the current cursor position. If the cursor is at home then the new line goes before the current line and is assigned a line number equal to the number of the preceding line (or 0 if there isn't one) plus 1. However if this would be greater than or equal to the current line number then the number assigned will be one less than the current line number.

If the cursor is not at home then the new text will be appended to the left hand part of the current line, and the right hand part will become a new line.

The line number assigned will be the current line number plus 1 or the following line number minus 1, whichever is the smaller.

Errors may result if the line numbers are already in conflict, otherwise only a warning will be given. Errors will also occur if the editor workspace is not large enough or the resulting line would be too long.

The cursor is always repositioned at the end of the inserted text; thus if the cursor was at home, it will remain at home, otherwise it will be repositioned immediately after the new line number.

5.7.3 L-SUB (DELETE LINES)

<n>,L<sub>

Delete <n> lines starting from current cursor position. If <n> is omitted then the comma must also be omitted and a value of 1 is assumed.

If cursor is at home then the current line is deleted and the following line becomes the new current line. This process is repeated <n> times or until the end of the file is reached.

If the cursor is not at home then the right part of the current line is deleted, the following line number is removed and the following line joined to the left part of the current line, provided that this does not result in too long a line. If this is the case then an error is given and nothing is deleted. The new line becomes the new current line.

It will be seen that the L<sub> command will reverse a <CR> (insert text) command if the cursor is repositioned to the start of the text that was inserted.

The not-at-home <n>,L<sub> command will repeat <n> times or until the end of file, or until a deletion can not be done due to a line being too long.

If the number specified is preceded by a "-" sign then lines will be deleted backwards from the current cursor position. Thus if the cursor is at home then -L<sub> will delete the previous line.

If the cursor is not at home then the left hand part of the current line and the current line number will be deleted, and the right part of the current line will be concatenated to the previous line, provided that the resulting line is not too long.

The cursor remains at the same position after a delete line operation.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7.4 ADD (INSERT STRING)

<string><add>

Insert <string> before current character position. Line length must not be exceeded.

5.7.5 SET (OVERWRITE STRING)

<string><set>

Overwrite characters in the current line starting at current character. If <string> is longer than the right part of the current line then the line will be extended, provided that this does not cause the line to exceed 79 characters.

If an <add> or a <set> command is executed while the cursor is at the end-of-file position then a new line number will be assigned (provided that there is sufficient buffer space) before the command is executed.

5.7.6 SUB (DELETE CHARACTERS)

<n><sub>

Delete <n> chars starting at current character position. cursor remains at the same position. This is a character command and will not delete lines. Thus <sub> will delete the rest of the current line. If n is omitted a value of 1 is assumed.

If the number specified is negative then characters will be deleted backwards starting with the previous character. (the line number will not be deleted)

All three of the preceeding commands will automatically move the cursor out of the home position if necessary before execution.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7.7 M (MUSIC KEYBOARD INPUT)

<+ or ->M<sharp or flat><default octave ><esc>

This command controls the inputting of characters from the music keyboard. On initial entry to the editor the music keyboard is disabled. After the "M" command is given, Switch 1 on the music keyboard will turn the music keyboard inputting on and off. When turned on, key depressions on the music keyboard will cause the M.C.L. representation of the note to be inserted into the file being edited at the current cursor position. The exact format of the note generated is determined by options specified on the M command, but is in the general format

:<note name><accidental (if any)><octave(if any)>

Note that no timing information is taken from the notes entered on the music keyboard.

Command format:

If a sharp or flat character follows immediately after the letter M then all accidentals will be written using that character. Thus you can select whether to write Bflat or Asharp. If no accidental is specified then the previous selection will remain, or "flat" if there has been no previous selection.

After the accidental, if any, a default octave may be specified in the form "O=<number>", where number must be from 0 to 7. If 0 then the default octave feature will be turned off and all octave values written in absolute form. Any other value will set the default octave and the octave value for keystrokes will be written as a relative value. This will facilitate later transposition if desired.

Examples of M command:

+M<esc> enable keyboard inputting. Attach the octave number to each note.

+M<ctrl F>O=4<esc> enable keyboard inputting. Use flats, make all octaves relative to octave 4.

As well as the above, switch number 2 on the music keyboard is designated as the "chord switch", to make entry of chords easier. When switch 2 is pressed an open bracket "(" is written into the input line, and when it is released a close bracket ")" is put into the file. Thus all keystrokes entered while the switch is depressed will be considered by the M.C.L. editor as a chord.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

To turn the keyboard off the command "-M<esc>" must be given. The keyboard can be temporarily turned off by releasing switch 1, or pressing and releasing if it is already up. The next depression will then turn the keyboard back on.

If the cursor is at the start of a line or if the input line already contains some user inserted characters then the colon (:) will be omitted.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7.8 L (LOCATE LINE NUMBER)

<n>L<esc>

Locate line number <n> or the next higher line number <n> defaults to 0000. Thus L<esc> will go to the start of the file and &L<esc> will go to the end of the file. 5.7.9 S (RESEQUENCE LINE NUMBERS)

<n1>,<n2>S<esc>

Resequence line numbers. This command is used to make room in the line numbers for more lines to be inserted. For this reason it works differently depending on whether the cursor is at home. If it is, then resequencing starts with the current line because new lines will be inserted before it. Otherwise resequencing starts with the following line.

<n1> is the increment to be applied between successive line numbers, and defaults to 10. <n2> is the starting value, which must not conflict with the previous line number (or current line number if not at home). If <n2> is not specified then it defaults to the previous line number (or current line number) plus the increment.

The maximum line number is 9999 and a warning will be given if this value is reached before end of file. If <n1> is omitted, then the comma may also be omitted.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

TYPE TWO COMMANDS (continued)

5.7.10 D (DISPLAY TEXT)

<n>D<esc>:

Display <n> lines on the screen starting at current line. after a display the screen will not be showing the current cursor position. to avoid silly mistakes the editor will not accept any cursor dependent commands but will give the message "NO LINE OPEN". To reopen the current line the command char <esc> must be typed on its own, after which the screen will be rewritten at the correct position.

If a negative number is specified before the D command then the display command will scroll downwards towards the start of the file.

5.7.11 P (PRINT TEXT ON LINE PRINTER)

<n>P<esc>

Print <n> lines on the line printer. The name of the currently open file is printed at the start of a listing and at the top of each page. Alternative forms of the print command are:

P/<esc> which will cause the printer to eject one page.

P:<text><esc> will print <text> on the line printer at the current page position then advance to the next line. Thus P:<esc> will skip 1 line.

If the printer is not connected or not ready (power off) when the P command is given, the error message "PRINTER NOT READY" will be displayed and the command aborted.

Both the display and print commands may be temporarily halted by pressing <control>W (for Wait), and resumed by pressing <control>W again. Alternatively the command may be aborted altogether by pressing <control><esc> (known as BREAK).

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7.12 F (FIND STRING)

<n>F<string><esc>

Search file from current position for <n> occurrences of the string <string>. Each time the string is found the cursor is repositioned to the START of the string, and it will not be moved until another instance is found. <n> defaults to 1. <string> must have at least one character.

5.7.13 C (CHANGE STRING)

<n>C<string1>\<string2><esc>

Search rest of file from current position and change <n> instances of <string1> to <string2>. Each time a string is changed the cursor is repositioned to the start of the string. <n> defaults to 1. <string1> must have at least 1 char. <string2> may be null. The delimiter "\" MUST occur exactly once in the command line.

The backslash character (\) is obtained by typing <shift>L on the Fairlight's alphanumeric keyboard.

5.7.14 \ (CHANGE PREVIOUSLY FOUND STRING)

\<string><esc>

This command enables a previously found or changed string to be changed again without retyping it. Whenever a string is found by the F or C commands its position and length are saved (until the cursor is moved again). This is referred to as the "current string" and it may be of any length including zero (after a CHANGE with null string2) If there is no current string an error will result.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7.15 R (READ TEXT INTO COPY BUFFER)

<n>R<esc>

Read <n> lines into the editor's copy buffer. This buffer is separate from the normal workspace and is used to save pieces of text for later insertion into the same file or a different file. Thus it can be used for block editing of files. Its contents will remain intact even after exiting the editor provided that no other command overlay is loaded.

The copy buffer can be opened for editing by the "B" command and the contents displayed, printed or changed, as for a complete file.

The read command reads from the current cursor position for <n> lines, <n> defaulting to 1. If the cursor is not at home then the first line read in will consist of the current line number and the right part of the current line. Only whole lines are stored in the buffer.

The size of the copy buffer is approximately 4K characters; if the number of lines specified in the R command will not fit into the copy buffer then the error message "COPY BUFFER TOO SMALL" will be given, and NO TEXT will be moved, so the contents of the buffer will remain unchanged.

5.7.16 W (WRITE COPY BUFFER INTO FILE)

<n>W<esc>

Write the contents of the copy buffer into the current file before cursor position <n> times. Only whole lines are inserted; thus if the cursor is not at home the current line will be split into two pieces before and after the inserted text. This is a line oriented command.

Line numbers are assigned as for multiple <carriage return> commands, so if a conflict occurs the last inserted lines will all have the same number and a resequence should be done. <n> defaults to 1.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7.17 B (OPEN COPY BUFFER)

B<esc>

Open the copy buffer for editing. The current file will become inaccessible but its cursor position will remain intact. While the copy buffer is open it may be edited using all the commands available except of course the P and W commands.

5.7.18 O (OPEN OR REOPEN FILE)

O<esc>

Close the copy buffer and reopen the current file at its previous cursor position. If the buffer is not open then do nothing

O<filename><esc>

Close the current file and open <filename>. If the copy buffer is open it will be closed first and the current file displayed to give a visual warning that what you were looking at wasn't the current file. The filename must refer to a file which is currently loaded in memory. Whenever a file is closed its name and cursor position is saved for use by the O- command.

All file opening commands will close the copy buffer if necessary before closing the current file.

O-<esc>

Open previous file, providing that there has been one. The current file now becomes the previous file. After the file is opened the cursor will be positioned to its previous location.

SECTION 5 - M.C.L. EDITOR (CONTINUED)

5.7.19 N (CREATE NEW FILE)

N<filename1>,<filename2><esc>

Close the current file and then create a new file called <filename1>. This file must not be loaded already and cannot be the same name as the current file. If <filename2> is specified (optional) then the new file will be initialised with the contents of <filename2>, which must be loaded as for the O commands.

The second filename may be replaced by the special symbol "=" which will then cause the new file to be initialized to the contents of the current file.

5.7.20 E (EXIT FROM EDITOR)

E<filename><esc>:

End edit, close the current file and return to the M.C.L. command interpreter. The message "END OF EDIT" will be displayed on the console. If <filename> is specified then the current file will be renamed before it is closed. <filename> must not be the name of a file currently loaded. If so then an error message will be displayed and the file will remain open.

If the file currently being edited contains no text at all then the empty file will be deleted and will not appear in the directory.

APPENDIX A - EXAMPLE SEQUENCE FILE

APPENDIX A - EXAMPLE SEQUENCE FILE

```
*
* THIS SEQUENCE FILE IS AN EXAMPLE OF MANY OF THE FEATURES OF
* THE FAIRLIGHT MUSIC COMPOSITION LANGUAGE.
*
* ( NOTE THAT THE SHARPS, FLATS AND NATURALS APPEAR AS
* s, f, and n WHEN PRINTED, ALTHOUGH THEY ARE
* DISPLAYED AS THE CORRECT SYMBOLS
* ON THE FAIRLIGHT C.R.T. GRAPHICS TERMINAL)
*
*
* DEFAULTS ASSUMED UPON STARTING TO PLAY AN MCL PIECE
!B=48: O=3: V=15: G=1/4: T=0
* BEAT=48, OCTAVE=3, VELOCITY=15, GAP TIME=1/4,
* TRANSPOSITION=0 (NONE)
F:G:A:fB:C:D:E:F3
* SCALE PLAYED, OCTAVE 2 FOR ALL EXCEPT ENING NOTE.

* DEFAULT IS SET BY AN "="
!B=32
*PLAY A SCALE AT BEAT=32
F:G:A:fB:C:D:E:F3

* DEFAULT OCTAVE SETTING
!O=3
* PLAY A SCALE AT THIS OCTAVE
F:G:A:fB:C:D:E:F4

* DEFAULT ACCIDENTAL
* MULTIPLE DEFAULT ACCIDENTALS SPECIFY A KEY SIGNATURE.
* EXAMPLE WOULD BE -- !fB:fE:fA
!fB
* PLAY A SCALE WITH THIS DEFAULT ACCIDENTAL SETTING
F:G:A:F:C:D:E:F4

* DEFAULT VELOCITY
!V=10
* PLAY A SCALE AT THIS VELOCITY
F:G:A:F:C:D:E:F4

* TRANSPOSE NOTES UP 5 SEMI-TONES
!T=5
* PLAY A SCALE AT THIS TRANSPOSITION
F:G:A:F:C:D:E:F4

* CHANGE "DEFAULT GAP TIME" FOR NOTES
!G=2/3
* DEFAULT TIME,
* NOTE WILL PLAY FOR 1/3 OF THE BEAT, AND THEN RELEASE FOR 2/3.
F:G:A:B:C:D:E:F4

* CHANGE FROM "GAP TIME" DEFAULT TO "HOLD TIME" DEFAULT.
```

APPENDIX A - EXAMPLE SEQUENCE FILE (CONTINUED)

* OR FIAT (CONTROL F) BEFORE OR AFTER THE NOTE.
* THIS OVERRIDES A DEFAULT ACCIDENTAL OR KEY SIGNATURE.
Fs:sG:An:Bn:Cn:Df:fE:Fs4

* RELATIVE OCTAVE AND VELOCITY
* PLACE A + OR - IN FRONT OF THE OCTAVE NUMBER,
* OR THE VELOCITY NUMBER.
FV+2:Gf+2V+4:A-1:B-:C+V-:sD-2V-3,2,1:E:F+
* NOTE THAT +1 AND -1 CAN BE WRITTEN WITHOUT THE 1.

* THEREFORE THE FORMAT OF A NOTE IS:
* (s/f/n)LETTER(s/f/n)(OCTAVE)(VELOCITY)(,TIME(,GAP TIME))

* SEPARATORS ARE OPTIONAL BETWEEN NOTES AND DEFAULT SETTINGS.
* THEY ARE ":" AND ";".
* A WORD OF CAUTION/WARNING:
* SINCE A ACCIDENTAL (s/f/n) MAY COME BEFORE OR AFTER A NOTE --
* THE LINE "F sG" WILL BE TAKEN AS Fs G.
* THEREFORE ,ALWAYS USE THE ":".
FGABCEDEF+

* CHORDS CAN BE SPECIFIED
* AS NOTES ENCLOSED INSIDE PARENTHESES "(" AND ")"
* THE TIME FOR A CHORD FOLLOWS THE RIGHT (CLOSING) PARENTHESIS.
(C:E:G):(D:F+:A),2,1: (E:G+:B)V13,2,1 (FAC)2,1
* BECAUSE IT IS SO OFTEN FORGOTTEN--
* THE "," AFTER THE) IS OPTIONAL,
* UNLESS A VELOCITY IS SPECIFIED.

* USE A CONTROL TO VARY THE VIBRATO DEPTH. NOTE THAT FOR
* THIS EXAMPLE TO WORK, CNTRL5 MUST BE ASSIGNED TO VIBRATO
* DEPTH ON PAGE 7.
* THE SCALE WILL BE PLAYED FOUR TIMES, THE FIRST TIME WITH
* NO VIBRATO, THEN WITH 10 UNITS MORE VIBRATO EACH REPEAT.
C5=0 <F:G:A:B:C:I:E:F+ C5=+10>4

* USE A SWITCH TO SELECTIVELY TURN PORTAMENTO ON AND OFF.
* NOTE THAT FOR THIS EXAMPLE TO WORK, SWTCH3 MUST BE
* ASSIGNED TO PORTAMENTO ON PAGE 7.
S3=0 F:G:A S3=1 B:C S3=0 D:E:F+

END

APPENDIX A - EXAMPLE SEQUENCE FILE (CONTINUED)

* OR FIAT (CONTROL F) BEFORE OR AFTER THE NOTE.
* THIS OVERRIDES A DEFAULT ACCIDENTAL OR KEY SIGNATURE.
Fs:sG:An:Bn:Cn:Df:fE:F+4

* RELATIVE OCTAVE AND VELOCITY
* PLACE A + OR - IN FRONT OF THE OCTAVE NUMBER,
* OR THE VELOCITY NUMBER.
FV+2:Cf+2V+4:A-1:B-:C+V-:sD-2V-3,2,1:E:F+
* NOTE THAT +1 AND -1 CAN BE WRITTEN WITHOUT THE 1.

* THEREFORE THE FORMAT OF A NOTE IS:
* (s/f/n)LETTER(s/f/n)(OCTAVE)(VELOCITY)(,TIME(,GAP TIME))

* SEPARATORS ARE OPTIONAL BETWEEN NOTES AND DEFAULT SETTINGS.
* THEY ARE ":" AND ";".
* A WORD OF CAUTION/WARNING:
* SINCE A ACCIDENTAL (s/f/n) MAY COME BEFORE OR AFTER A NOTE --
* THE LINE "F sG" WILL BE TAKEN AS Fs G.
* THEREFORE ,ALWAYS USE THE ":".
FGABCEDF+

* CHORDS CAN BE SPECIFIED
* AS NOTES ENCLOSED INSIDE PARENTHESES "(" AND ")"
* THE TIME FOR A CHORD FOLLOWS THE RIGHT (CLOSING) PARENTHESIS.
(C:E:G):(D:F+:A),2,1: (E:G+:B)V13,2,1 (FAC)2,1
* BECAUSE IT IS SO OFTEN FORGOTTEN--
* THE "," AFTER THE) IS OPTIONAL,
* UNLESS A VELOCITY IS SPECIFIED.

* USE A CONTROL TO VARY THE VIBRATO DEPTH. NOTE THAT FOR
* THIS EXAMPLE TO WORK, CNTRL5 MUST BE ASSIGNED TO VIBRATO
* DEPTH ON PAGE 7.
* THE SCALE WILL BE PLAYED FOUR TIMES, THE FIRST TIME WITH
* NO VIBRATO, THEN WITH 10 UNITS MORE VIBRATO EACH REPEAT.
C5=0 <F:G:A:B:C:I:E:F+ C5=+10>4

* USE A SWITCH TO SELECTIVELY TURN PORTAMENTO ON AND OFF.
* NOTE THAT FOR THIS EXAMPLE TO WORK, SWTCH3 MUST BE
* ASSIGNED TO PORTAMENTO ON PAGE 7.
S3=0 F:G:A S3=1 B:C S3=0 D:E:F+

END

APPENDIX B - A SIMPLE EXAMPLE (CONTINUED)

APPENDIX B - A SIMPLE EXAMPLE

Reproduced below is an example of the M.C.L. version of "Three Blind Mice". In this instance, three parts are specified which play the melody in canon using three different keyboards, and hence three different voices may be assigned to the parts.

The steps involved in generating this M.C.L. piece might be:

- 1) Set up PAGE 3 with three different registers, each with a different voice, and assign keyboards 1 to 3 to registers A to C respectively.

- 2) Select PAGE C to enter the M.C.L. sub-system, and use the EDITOR to create a sequence file. The command would be: NEW,MOUSE.SS.

- 3) Once the edit has been ended (using the E<esc> command), the message END OF EDIT will be displayed. The command PLAY MOUSE.SS will then play the sequence just created, and as no PART has been specified so far, it will default to keyboard 1.

- 4) Create the three parts desired, in this case MOUSEA.PT, MOUSEB.PT and MOUSEC.PT. In order to make the MOUSEB and MOUSEC parts wait the correct time before starting, a sequence consisting of just rests is introduced into their PART files. Remember to start the line of sequence with a " or the M.C.L. will look for a sequence called B=24 .

- 5) Now create the REST8.SS sequence.

- 6) Create the PIECE, MICE.PC which is simply a list of the PARTS to be played. The order in which the parts are specified is of no consequence in this case as the keyboard number has been specified for each PART. Otherwise the order would dictate which part plays which keyboard.

- 7) Play the PIECE, using the command: PLAY,MOUSE.PC
Note any mistakes in the music or error messages which may be displayed.

- 8) If your C.M.I. is equipped with a Line Printer, you can list the piece for reference and as an aid to debugging. The command:
PRINT ((MICE.PC will give you a listing of the PIECE, PARTS and SEQUENCES like the one below.

- 9) Use the editor to correct any mistakes.

- 10) Don't forget to SAVE your work on disk once you are satisfied with it. Use the command: SAVE,((MOUSE.PC to save the piece, parts, and sequences.

APPENDIX B - A SIMPLE EXAMPLE (CONTINUED)

FAIRLIGHT COMPOSER - LEVEL 5.3

MICE.PC

PAGE 1

FILENAME = MICE .PC

```
0001 * THIS IS THE "THREE BLIND MICE" PIECE.
0002 MOUSEA
0003 MOUSEB
0004 MOUSEC
0005 END
```

FILENAME = MCUSEA .PT

```
0001 IK=1
0002 MOUSE
0003 END
```

FILENAME = MOUSE .SS

```
0001 * THIS IS THE CENTRAL SEQUENCE TO BE
0002 * PLAYED BY ALL THREE PARTS IN CANON.
0003 <
0004 IB=24:0=3
0005 <E:D:C:R>2
0006 G+:F+,1/2:F+,1/2:E:R G+:F+,1/2:F+,1/2:E:R,2/3
0025 * NOW FOR THE MIDDLE BIT
0035 IO=+:B=8
0045 <G:C,2:C:B:A:B:C,2:G:G:R>3 F
0055 *THE LAST BAR
0065 O=-:B=12
0075 E:D:C:R
0085 * NOW REPEAT THE WHOLE LOT THREE TIMES
0095 >3
0105 IND
```

FILENAME = MOUSEB .PT

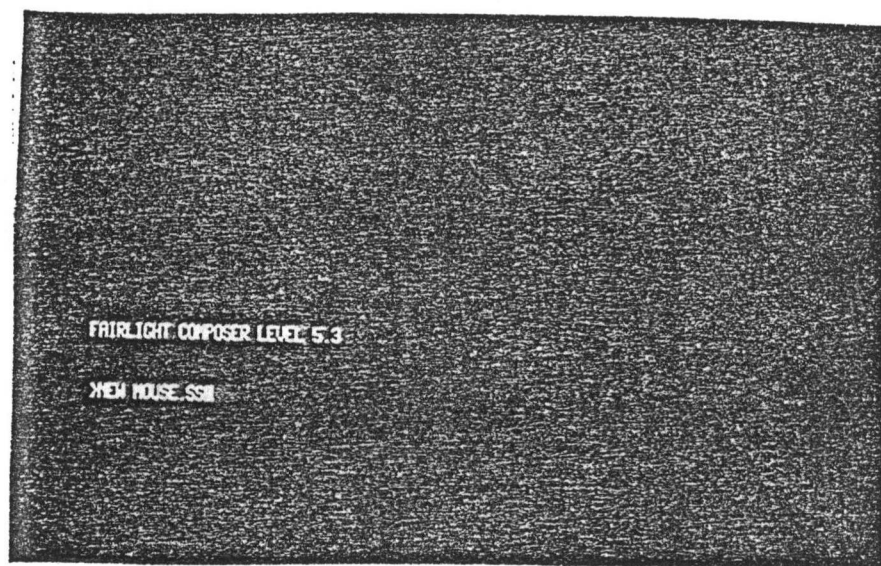
```
0001 * SECOND PART OF THE ROUND.
0002 * PLAY ON KEYBOARD 2
0003 IK=2
0004 * REST FOR 8 BEATS
0005 "B=24:R,8
0006 MOUSE
0007 END
```

APPENDIX B - A SIMPLE EXAMPLE (CONTINUED)

FILENAME = MOUSEC .PT

```
0010 * THIRD PART OF THE ROUND.  
0020 * PLAY ON KEYBOARD 3  
0040 !K=3  
0050 * NOW REST FOR 16 BEATS  
0060 "B=24:R,16  
0070 MCOUSE  
0080 END
```

This is how the screen looks after entering Page C and issuing the command: NEW MCOUSE.SS (to create a new sequence). As soon as RETURN is hit the editor will be invoked (see photos next page).



APPENDIX B - A SIMPLE EXAMPLE (CONTINUED)

This is how the EDITOR looks. The second line from the top is where commands are entered. The empty box towards the middle of the screen is where the new text is about to be inserted. The line numbers are generated automatically (the next line will be number 36).

```

FILE IO - COMPOSED EDITOR - LEVEL 1 4 - READY FILENAME = MOUSE 00
0001 * THIS IS THE CENTRAL SEQUENCE TO BE
0002 * PLAYED BY ALL THREE PARTS IN CANON.
0003 C
0004 !B=24:0-3
0005 !E:D:C:R:2
0006 C:F:1/2:F:1/2:E:R G:F:1/2:F:1/2:E:R:2/3
0025 * NOW FOR THE MIDDLE BIT
0035 !O=:B=8

```

In this photograph the cursor is positioned over a typing mistake (the "O" in ELOND). An "I" has been entered into the command line (second line from the top). If the SET key is now struck, the "O" will be replaced by an "I".

```

FILE IO - COMPOSED EDITOR - LEVEL 1 4 - READY FILENAME = MOUSE 00
0001 * THIS IS THE "THREE BLOND NICE PIECE"
0002 MOUSER
0003 MOUSER
0004 MOUSER
0005 END

```


APPENDIX B - A SIMPLE EXAMPLE (CONTINUED)

Once editing has been ended (using the E<esc> command) the screen is cleared and the message:
MICE.PC END OF EDIT is displayed. The SAVE command has then been used. The "(" before the filename has caused all the parts and sequences to be saved as well as the PIECE.

```
MICE PC END OF EDIT
)SAVE ((MICE.PC
SAVING MICE PC
SAVING MUSER PT
SAVING MUSER MOUSE SS
SAVING MUSER PT
SAVING MUSER PT
*)
```

The LOAD command has been issued. The "(" before the filename has caused all the parts and sequences to be loaded as well. Note that the "))" is optional.

```
FAIRLIGHT COMPOSER LEVEL 5.3
)LOAD ((MICE.PC)
LOADING MICE PC
LOADING MUSER PT
LOADING MUSER MOUSE SS
LOADING MUSER PT
LOADING MUSER PT
)PLAY MICE.PC
*)
```


APPENDIX C - EXTERNAL SYNCHRONISATION

- 1) Connect an audio oscillator to the tape recorder input associated with the track which is to carry the SYNC tone. Any oscillator with a variable frequency output in the range 100 to 5000 Hz can be used. The shape of the waveform is irrelevant.
- 2) Connect the appropriate output of the tape recorder to the SYNC input of the C.M.I. (Pin 2 of the SYNC connector). This is a single-ended (unbalanced) input, requiring a minimum level of 1 volt P-P for reliable operation.
- 3) The M.C.L. will play at the speed determined by the precise frequency of the SYNC tone. To find out the correct frequency, put the tape recorder into "bypass" mode so that the signal from the oscillator will pass straight through to the output (and hence to the C.M.I. SYNC input). Then select external sync (using the command: SYNC=EXT) and play the M.C.L. piece. Adjusting the frequency of the oscillator will vary the replay speed of the M.C.L. Select a suitable speed or range of speeds.
- 4) Lay down the SYNC track. The M.C.L. will start playing as soon as the tone starts coming off tape, so make sure that the start of the SYNC tone is clean and is preceded by a few seconds of silence.

It is possible to vary the speed of the piece dynamically by varying the oscillator frequency while laying down the SYNC track. For this purpose it is necessary to be playing the M.C.L. piece in external sync mode while laying down the sync track.

Let the SYNC track run for a few seconds longer than the M.C.L. piece.
- 5) From now on, all M.C.L. playing will faithfully follow this SYNC track (unless SYNC=INT is re-selected). Lay down each group of M.C.L. parts on a separate tape track, making sure that the PLAY is executed with SYNC=EXT.

APPENDIX C - EXTERNAL SYNCHRONISATION

APPENDIX C - EXTERNAL SYNCHRONISATION

The C.M.I. is provided with a three pin CANNON connector labelled SYNC.

The function of each pin is:

| | |
|-------|---------------------|
| PIN 1 | SIGNAL GROUND |
| PIN 2 | EXTERNAL SYNC INPUT |
| PIN 3 | CLICK OUTPUT |

USING CLICK OUTPUT

This is a pulse-type output which is used primarily for a beat reference for live musician accompaniment. It is also useful for debugging purposes.

The click output can be connected to an audio amplifier via any single-ended line-level input. The amplitude of the click is about 1 volt P-P. The click can also be heard via the C.M.I. monitor speaker output. The volume of the click is controlled by the SYNC MONITOR control on the rear panel of the C.M.I.

The relationship between the click output and the beat rate of the M.C.L. piece is set by the command:

CLICK=r,i

where "r" is the number of clock ticks per click and "i" is the number of clicks introduction that will be heard before the piece starts playing. Because the relationship between clock ticks and the beat rate of the M.C.L. is set by the B=n statement in a sequence or part file, using the same number for B=n and CLICK=n will result in one click per beat. For example, if the M.C.L. uses B=24, then CLICK=96,4 will result in one click per 4-beat bar and 4 clicks (=4 bars) intro.

SYNCHRONISING TO MULTI-TRACK TAPE

If the complete M.C.L. piece requires more than the allowable maximum of eight parts, multi-track tape can be used to allow further parts to be laid down. In this case it is necessary to lay down a sync track on an unused track of the tape so that the M.C.L. can follow variations in the tape speed for perfect synchronisation. The procedure is as follows:

964-8006 → Long

Ron Allen → 656 2575